

RRRRRRRRRRRR	MM	MM	SSSSSSSSSSSS
RRRRRRRRRRRR	MM	MM	SSSSSSSSSSSS
RRRRRRRRRRRR	MM	MM	SSSSSSSSSSSS
RRR RRR	MMMM	MMMM	SSS
RRR RRR	MMMM	MMMM	SSS
RRR RRR	MMMM	MMMM	SSS
RRR RRR	MM	MM	SSS
RRR RRR	MM	MM	SSS
RRR RRR	MM	MM	SSS
RRRRRRRRRRRR	MM	MM	SSSSSSSS
RRRRRRRRRRRR	MM	MM	SSSSSSSS
RRRRRRRRRRRR	MM	MM	SSSSSSSS
RRR RRR	MM	MM	SSS
RRR RRR	MM	MM	SSS
RRR RRR	MM	MM	SSS
RRR RRR	MM	MM	SSS
RRR RRR	MM	MM	SSS
RRR RRR	MM	MM	SSS
RRR RRR	MM	MM	SSSSSSSSSS
RRR RRR	MM	MM	SSSSSSSSSS
RRR RRR	MM	MM	SSSSSSSSSS

RRRRRRRR	MM	MM	SSSSSSSS	000000	RRRRRRRR	NN	NN	DDDDDDDD	WW	WW	NN	NN
RRRRRRRR	MM	MM	SSSSSSSS	000000	RRRRRRRR	NN	NN	DDDDDDDD	WW	WW	NN	NN
RR	RR	MMMM	MMMM	SS	00	RR	RR	NN	DD	DD	WW	WW
RR	RR	MMMM	MMMM	SS	00	RR	RR	NN	DD	DD	WW	WW
RR	RR	MM	MM	SS	00	RR	RR	NNNN	NN	DD	WW	WW
RR	RR	MM	MM	SS	00	RR	RR	NNNN	NN	DD	WW	WW
RRRRRRRR	MM	MM	SSSSSS	00 00 00	RRRRRRRR	NN	NN	NN	DD	DD	WW	WW
RRRRRRRR	MM	MM	SSSSSS	00 00 00	RRRRRRRR	NN	NN	NN	DD	DD	WW	WW
RR	RR	MM	MM	SS	0000	RR	RR	NNNN	DD	DD	WW	WW
RR	RR	MM	MM	SS	0000	RR	RR	NNNN	DD	DD	WW	WW
RR	RR	MM	MM	SS	00	RR	RR	NN	NN	DD	WWWW	WWWW
RR	RR	MM	MM	SS	00	RR	RR	NN	NN	DD	WWWW	WWWW
RR	RR	MM	MM	SSSSSSSS	000000	RR	RR	NN	DDDDDDDD	WW	WW	NN
RR	RR	MM	MM	SSSSSSSS	000000	RR	RR	NN	DDDDDDDD	WW	WW	NN

LL		SSSSSSSS
LL		SSSSSSSS
LL		SS
LL		SS
LL		SS
LL		SSSSSS
LL		SSSSSS
LL		SS
LL		SS
LL		SS
LLLLLLLL		SSSSSSSS
LLLLLLLL		SSSSSSSS

(2) 70
(3) 102

DECLARATIONS
RMS\$RMSRUNDWN - RMS I/O RUN DOWN

```
0000 1 $BEGIN RMSORNDWN.001,RMSRMS,<RMS IO RUN DOWN>
0000 2
0000 3 :*****
0000 4 :*****
0000 5 :*
0000 6 :* COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0000 7 :* DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0000 8 :* ALL RIGHTS RESERVED.
0000 9 :*
0000 10 :* THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0000 11 :* ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0000 12 :* INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0000 13 :* COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0000 14 :* OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0000 15 :* TRANSFERRED.
0000 16 :*
0000 17 :* THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0000 18 :* AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0000 19 :* CORPORATION.
0000 20 :*
0000 21 :* DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0000 22 :* SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0000 23 :*
0000 24 :*
0000 25 :*****
0000 26 :*
0000 27 :++
0000 28 : Facility: rms32
0000 29 :*
0000 30 : Abstract: this module insures all rms i/o activity is complete,
0000 31 : closes all files, and resets the ifab and irab tables.
0000 32 :*
0000 33 : Environment:
0000 34 : star processor running starlet exec.
0000 35 :*
0000 36 : Author: l f laverdure, creation date: 5-5-77
0000 37 :*
0000 38 : Modified By:
0000 39 :*
0000 40 : V04-001 RAS0332 Ron Schaefer 14-Sep-1984
0000 41 : ALWAYS re-enable ASTS when stalling inside rundown
0000 42 : as they could get disabled by the previous exec mode
0000 43 : thread of RMS that will never continue and re-enable them.
0000 44 :*
0000 45 : V03-005 DGB0040 Donald G. Blair 02-May-1984
0000 46 : If the PIO$V_INHAST bit is set when we start an
0000 47 : RMS operation, we conclude that the caller must be
0000 48 : at exec AST level or higher and would break RMS
0000 49 : synchronization rules if he were allowed to continue.
0000 50 : Return error. This fix also includes a change from
0000 51 : Jim Johnson to clear the FID correctly in GETDVIFID.
0000 52 :*
0000 53 : V03-004 SHZ0001 Stephen H. Zalewski 14-Sep-1983
0000 54 : Move routine RMSGETDVIFID from module RMOGETDVI to here, and
0000 55 : rename it GETDVIFID. Module RMOGETDVI has been evaporated.
0000 56 :*
0000 57 : V03-003 JWH0107 Jeffrey W. Horn 24-Sep-1982
```

RMSORNDWN
V04-001

RMS IO RUN DOWN

H 13

16-SEP-1984 01:29:13 VAX/VMS Macro V04-00
14-SEP-1984 22:32:57 [RMS.SRC]RMSORNDWN.MAR;2

Page 2
(1)

0000	58	:	Add call to RMSRU_UNLOCK to release locks held for the duration of a recovery unit.
0000	59	:	
0000	60	:	
0000	61	:	V03-002 KBT0316 Keith B. Thompson 8-Sep-1982
0000	62	:	Remove all S0 sharing code
0000	63	:	
0000	64	:	V03-001 KBT0191 Keith B. Thompson 23-Aug-1982
0000	65	:	Reorganize psects and rename entry points to single '\$'
0000	66	:	
0000	67	--	
0000	68		

RMS
V04

```
0000 70 .SBTTL DECLARATIONS
0000 71
0000 72 : Include Files:
0000 73 :
0000 74 :
0000 75
0000 76 :
0000 77 : Macros:
0000 78 :
0000 79
0000 80     $DEVDEF
0000 81     $FABDEF
0000 82     $FIBDEF
0000 83     $FWADEF
0000 84     $IFBDEF
0000 85     $IRBDEF
0000 86     $IMPDEF
0000 87     $SNWADef
0000 88     $PIODEF
0000 89     $PSLDEF
0000 90     $RLBDEF
0000 91     $RMSDEF
0000 92
0000 93 : Equated Symbols:
0000 94 :
0000 95 :
0000 96
0000 97 :
0000 98 : Own Storage:
0000 99 :
0000 100
```

0000 102 .SBTTL RMSS\$RMSRUNDW - RMS I/O RUN DOWN
0000 103
0000 104 :++
0000 105 : RMSS\$RMSRUNDW - RMS I/O run down
0000 106 :
0000 107 : this routine first determines the type of rundown desired, based
0000 108 : upon the second argument. if the type is "abort rms i/o", a branch
0000 109 : is made to rm\$last_chance, otherwise the routine checks that all ifabs and irabs
0000 110 : are inactive. if any found active this routine awaits their completion after
0000 111 : first performing a \$cancel i/o if not a file-oriented device.
0000 112 : when all i/o activity for the file is complete, \$close is
0000 113 : performed for the file. if the close failed for an output file
0000 114 : on a files-oriented device, an error is returned to the caller
0000 115 : who should note the error and recall this routine to run down
0000 116 : further files. if all files are successfully run down the
0000 117 : image ifab & irab tables are reset and return is made to the
0000 118 : caller with a success code.
0000 119 :
0000 120 : files are run down in this order:
0000 121 :
0000 122 : 1. indirect process permanent files
0000 123 : ('error' should be first)
0000 124 : 2. image files
0000 125 : 3. (only if caller's mode is not user and arg2=1)
0000 126 : process permanent files
0000 127 :
0000 128 : Calling sequence:
0000 129 : calls #2, sys\$rmsrundw
0000 130 :
0000 131 : Input Parameters:
0000 132 :
0000 133 : ap users argument list (2 arguments)
0000 134 :
0000 135 : arg1 descriptor for 22-character buffer
0000 136 : to receive information about
0000 137 : unsuccessfully closed output file
0000 138 : (device id and file id)
0000 139 :
0000 140 : arg2 rundown type, as follows:
0000 141 :
0000 142 : 0 - run down of image and indirect i/o for process permanent files
0000 143 : 1 - run down of image and process permanent files
0000 144 : (caller's mode must be other than user)
0000 145 : 2 - abort rms i/o (caller's mode must be exec or kernel)
0000 146 :
0000 147 : all others are reserved, but currently behave as type 0
0000 148 :
0000 149 : Implicit Inputs:
0000 150 :
0000 151 : caller's mode.
0000 152 :
0000 153 : Output Parameters:
0000 154 :
0000 155 : r0 status code
0000 156 : r1 destroyed
0000 157 :
0000 158 : Implicit Outputs:

0000 159 :
0000 160 : information describing an output file unsuccessfully closed is
0000 161 : stored in the caller-provided buffer in exactly the same
0000 162 : format as the dvi, did, and fid fields of the nam block.
0000 163 :
0000 164 : Completion Codes:
0000 165 :
0000 166 : standard rms, in particular:
0000 167 :
0000 168 : rms\$\$_suc - all files closed
0000 169 : rms\$\$_ccf - an output file could not be closed
0000 170 : successfully. caller-provided buffer
0000 171 : has information identifying the file
0000 172 : rms\$\$_ial - same as rms\$\$_ccf except could not
0000 173 : access caller's buffer to store file
0000 174 : id information.
0000 175 :
0000 176 : Side Effects:
0000 177 :
0000 178 : runs synchronously in exec mode inhibiting
0000 179 : and enabling asts as required.
0000 180 :
0000 181 :--
0000 182 :

**F

<pre> 58 DC 0000 184 SENTRY RMSS\$RMSRUNDWN 58 EF 0000 185 \$TSTPT RUNDWN 16 EF 0006 186 MOVPSL R11 02 08 000A 187 EXTZV #PSL\$V_PRVMOD,- 42 13 000D 188 #PSL\$S_PRVMOD,R11,R7 : save caller's mode 00 E2 0011 189 CMPL 8(AP),#2 : abort rms i/o? 40 00000000'9F 0013 190 BEQL RMSABORT : branch if yes 00 E2 0013 191 BBSS #PIO\$V_INHAST,- : br if RMS already in progress 0015 192 @#PIO\$GW_STATUS,ERRBUSY 001B 193 001B 194 : start by releasing locks held for the duration of a recovery unit, 001B 195 : if any. 001B 196 : 001B 197 : 001B 198 FFE2' 30 001B 199 BSBW RMSRU_UNLOCK 001E 200 001E 201 : 001E 202 : next run down indirect i/o on process-permanent files 001E 203 : 001E 204 58 00000000'9F DE 001E 205 MOVAL @#PIO\$GW_PIOIMPA,R11 : get pio impure area address 58 01 D0 0025 206 ASSUME IMP\$W_RMSSSTATUS EQ 0 006B 30 0025 207 MOVL #1,R8 : indicate indirect run down 0028 208 BSBW RUNDWN : do the run down 0028 209 0028 210 : (note: clears r8) 0028 211 : now run down the image 0028 212 : 0028 213 58 0000'CB DE 0028 214 MOVAL W^PIO\$GW_IIOIMPA-PIO\$GW_PIOIMPA(R11),R11 0030 215 0030 216 : point to image impure area 0030 217 : 0030 218 : 0030 219 0063 30 0030 220 BSBW RUNDWN : do the run down 18 BB D4 0033 221 CLRL @IMP\$L_IFABTBL(R11) : reset ifab table link 1C BB D4 0036 222 CLRL @IMP\$L_IRABTBL(R11) : reset irab table link 0039 223 0039 224 : 0039 225 : point to process 0039 226 : i/o impure area again 0039 227 : 0039 228 58 0000'CB DE 0039 229 MOVAL W^PIO\$GW_PIOIMPA-PIO\$GW_IIOIMPA(R11),R11 003E 230 : 003E 231 : At this point there used to be code to return any whole pages 003E 232 : on the FMLH free space list back to the process i/o free page 003E 233 : list. The space on the FMLH list is currently (v 2) used only 003E 234 : for ASB allocation on IFAB operations and will bugcheck if space 003E 235 : is not found. The behavior is now that a page will be added to 003E 236 : the FMLH list the first time a process stalls on an IFAB operation 003E 237 : and will remain there for the life of the process. 003E 238 : 003E 239 : 003E 240 : </pre>
--

003E 241 : now run down direct i/o on process-permanent files if desired
003E 242 :
003E 243 :
01 08 AC 91 003E 244 CMPB 8(AP),#1 : ppf rundown?
07 12 0042 245 BNEQ XITSUC : branch if not
03 57 91 0044 246 CMPB R7,#PSL\$C_USER : caller sufficiently privileged?
02 13 0047 247 BEQL XITSUC : branch if not
4B 10 0049 248 60\$: BSB8 RUNDWN : do the run down
004B 249 XITSUC: RMSSUC :
004E 250 EXIT: :
11 10 0052 251 SSB #16, R0 : stamp 'rms' on status code
04 0054 252 BSB8 ENBAST : enable asts
0055 253 RET : back to caller
0055 254 :
0055 255 : branch to rm\$last_chance to do async process deletion rms i/o abort
0055 256 :
0055 257 :
0055 258 :
0055 259 RMSABORT: :
00000000'EF 17 0055 260 JMP RM\$LAST_CHANCE

```

005B 262 :
005B 263 : If the PIO$V_INHAST bit is already set, we
005B 264 : conclude that the caller must be at exec ast level or higher
005B 265 : (otherwise, he could not have kicked off an RMS operation
005B 266 : while RMS was already in progress) and would break RMS
005B 267 : synchronization rules if allowed to continue. Return RMSS_BUSY
005B 268 : status when this happens.
005B 269 :
005B 270 :
005B 271 ERRBUSY:
005B 272     RMSERR BUSY
0060 273     SSB #16,RO
04   0064 274     RET
0065 275 :
0065 276 : enable rms ast's, reenabling exec ast's in all cases.
0065 277 :
0065 278 :
0065 279 ENBAST: CSB #PIO$V_INHAST, @#PIO$GW_STATUS
006D 280 :
006D 281 :
006D 282 : clear ast inhibit and enable asts
006D 283 :
006D 284 :
05   006D 285 :
006D 286     $SETAST_S #1 ; enable exec mode asts
0076 287     RSB
0077 288 :
0077 289 :
0077 290 : inhibit rms asts
0077 291 :
0077 292 :
00000000'9F 01 A8 0077 293 INHAST: BISW2 #1@PIO$V_INHAST, @#PIO$GW_STATUS
05   007E 294     RSB
007F 295 :
007F 296 :
007F 297 : wait for rms operation completion
007F 298 :
007F 299 :
DB   10 0088 300 WAIT: $CLREF_S #IMPSC_IOREFN : clear rms event flag
          BSB# ENBAST : enable asts
          $WAITFR_S #IMPSC_IOREFN : wait for flag
          BSB# INHAST : re-inhibit asts
          RSB
E2   10 008A 301 :
          302 :
          303 :
          304 :
          305 :
          306 :
          307 :
          308 :
          309 :
          310 :
          311 :
          312 :
          313 :
          314 :
          315 :
          316 :
          317 :
          318 :
          319 :
          320 :
          321 :
          322 :
          323 :
          324 :
          325 :
          326 :
          327 :
          328 :
          329 :
          330 :
          331 :
          332 :
          333 :
          334 :
          335 :
          336 :
          337 :
          338 :
          339 :
          340 :
          341 :
          342 :
          343 :
          344 :
          345 :
          346 :
          347 :
          348 :
          349 :
          350 :
          351 :
          352 :
          353 :
          354 :
          355 :
          356 :
          357 :
          358 :
          359 :
          360 :
          361 :
          362 :
          363 :
          364 :
          365 :
          366 :
          367 :
          368 :
          369 :
          370 :
          371 :
          372 :
          373 :
          374 :
          375 :
          376 :
          377 :
          378 :
          379 :
          380 :
          381 :
          382 :
          383 :
          384 :
          385 :
          386 :
          387 :
          388 :
          389 :
          390 :
          391 :
          392 :
          393 :
          394 :
          395 :
          396 :
          397 :
          398 :
          399 :
          3100:
          3101:
          3102:
          3103:
          3104:
          3105:
          3106:
          3107:
          3108:
          3109:
          3110:
          3111:
          3112:
          3113:
          3114:
          3115:
          3116:
          3117:
          3118:
          3119:
          3120:
          3121:
          3122:
          3123:
          3124:
          3125:
          3126:
          3127:
          3128:
          3129:
          3130:
          3131:
          3132:
          3133:
          3134:
          3135:
          3136:
          3137:
          3138:
          3139:
          3140:
          3141:
          3142:
          3143:
          3144:
          3145:
          3146:
          3147:
          3148:
          3149:
          3150:
          3151:
          3152:
          3153:
          3154:
          3155:
          3156:
          3157:
          3158:
          3159:
          3160:
          3161:
          3162:
          3163:
          3164:
          3165:
          3166:
          3167:
          3168:
          3169:
          3170:
          3171:
          3172:
          3173:
          3174:
          3175:
          3176:
          3177:
          3178:
          3179:
          3180:
          3181:
          3182:
          3183:
          3184:
          3185:
          3186:
          3187:
          3188:
          3189:
          3190:
          3191:
          3192:
          3193:
          3194:
          3195:
          3196:
          3197:
          3198:
          3199:
          31100:
          31101:
          31102:
          31103:
          31104:
          31105:
          31106:
          31107:
          31108:
          31109:
          31110:
          31111:
          31112:
          31113:
          31114:
          31115:
          31116:
          31117:
          31118:
          31119:
          31120:
          31121:
          31122:
          31123:
          31124:
          31125:
          31126:
          31127:
          31128:
          31129:
          31130:
          31131:
          31132:
          31133:
          31134:
          31135:
          31136:
          31137:
          31138:
          31139:
          31140:
          31141:
          31142:
          31143:
          31144:
          31145:
          31146:
          31147:
          31148:
          31149:
          31150:
          31151:
          31152:
          31153:
          31154:
          31155:
          31156:
          31157:
          31158:
          31159:
          31160:
          31161:
          31162:
          31163:
          31164:
          31165:
          31166:
          31167:
          31168:
          31169:
          31170:
          31171:
          31172:
          31173:
          31174:
          31175:
          31176:
          31177:
          31178:
          31179:
          31180:
          31181:
          31182:
          31183:
          31184:
          31185:
          31186:
          31187:
          31188:
          31189:
          31190:
          31191:
          31192:
          31193:
          31194:
          31195:
          31196:
          31197:
          31198:
          31199:
          31200:
          31201:
          31202:
          31203:
          31204:
          31205:
          31206:
          31207:
          31208:
          31209:
          31210:
          31211:
          31212:
          31213:
          31214:
          31215:
          31216:
          31217:
          31218:
          31219:
          31220:
          31221:
          31222:
          31223:
          31224:
          31225:
          31226:
          31227:
          31228:
          31229:
          31230:
          31231:
          31232:
          31233:
          31234:
          31235:
          31236:
          31237:
          31238:
          31239:
          31240:
          31241:
          31242:
          31243:
          31244:
          31245:
          31246:
          31247:
          31248:
          31249:
          31250:
          31251:
          31252:
          31253:
          31254:
          31255:
          31256:
          31257:
          31258:
          31259:
          31260:
          31261:
          31262:
          31263:
          31264:
          31265:
          31266:
          31267:
          31268:
          31269:
          31270:
          31271:
          31272:
          31273:
          31274:
          31275:
          31276:
          31277:
          31278:
          31279:
          31280:
          31281:
          31282:
          31283:
          31284:
          31285:
          31286:
          31287:
          31288:
          31289:
          31290:
          31291:
          31292:
          31293:
          31294:
          31295:
          31296:
          31297:
          31298:
          31299:
          31300:
          31301:
          31302:
          31303:
          31304:
          31305:
          31306:
          31307:
          31308:
          31309:
          31310:
          31311:
          31312:
          31313:
          31314:
          31315:
          31316:
          31317:
          31318:
          31319:
          31320:
          31321:
          31322:
          31323:
          31324:
          31325:
          31326:
          31327:
          31328:
          31329:
          31330:
          31331:
          31332:
          31333:
          31334:
          31335:
          31336:
          31337:
          31338:
          31339:
          31340:
          31341:
          31342:
          31343:
          31344:
          31345:
          31346:
          31347:
          31348:
          31349:
          31350:
          31351:
          31352:
          31353:
          31354:
          31355:
          31356:
          31357:
          31358:
          31359:
          31360:
          31361:
          31362:
          31363:
          31364:
          31365:
          31366:
          31367:
          31368:
          31369:
          31370:
          31371:
          31372:
          31373:
          31374:
          31375:
          31376:
          31377:
          31378:
          31379:
          31380:
          31381:
          31382:
          31383:
          31384:
          31385:
          31386:
          31387:
          31388:
          31389:
          31390:
          31391:
          31392:
          31393:
          31394:
          31395:
          31396:
          31397:
          31398:
          31399:
          31400:
          31401:
          31402:
          31403:
          31404:
          31405:
          31406:
          31407:
          31408:
          31409:
          31410:
          31411:
          31412:
          31413:
          31414:
          31415:
          31416:
          31417:
          31418:
          31419:
          31420:
          31421:
          31422:
          31423:
          31424:
          31425:
          31426:
          31427:
          31428:
          31429:
          31430:
          31431:
          31432:
          31433:
          31434:
          31435:
          31436:
          31437:
          31438:
          31439:
          31440:
          31441:
          31442:
          31443:
          31444:
          31445:
          31446:
          31447:
          31448:
          31449:
          31450:
          31451:
          31452:
          31453:
          31454:
          31455:
          31456:
          31457:
          31458:
          31459:
          31460:
          31461:
          31462:
          31463:
          31464:
          31465:
          31466:
          31467:
          31468:
          31469:
          31470:
          31471:
          31472:
          31473:
          31474:
          31475:
          31476:
          31477:
          31478:
          31479:
          31480:
          31481:
          31482:
          31483:
          31484:
          31485:
          31486:
          31487:
          31488:
          31489:
          31490:
          31491:
          31492:
          31493:
          31494:
          31495:
          31496:
          31497:
          31498:
          31499:
          31500:
          31501:
          31502:
          31503:
          31504:
          31505:
          31506:
          31507:
          31508:
          31509:
          31510:
          31511:
          31512:
          31513:
          31514:
          31515:
          31516:
          31517:
          31518:
          31519:
          31520:
          31521:
          31522:
          31523:
          31524:
          31525:
          31526:
          31527:
          31528:
          31529:
          31530:
          31531:
          31532:
          31533:
          31534:
          31535:
          31536:
          31537:
          31538:
          31539:
          31540:
          31541:
          31542:
          31543:
          31544:
          31545:
          31546:
          31547:
          31548:
          31549:
          31550:
          31551:
          31552:
          31553:
          31554:
          31555:
          31556:
          31557:
          31558:
          31559:
          31560:
          31561:
          31562:
          31563:
          31564:
          31565:
          31566:
          31567:
          31568:
          31569:
          31570:
          31571:
          31572:
          31573:
          31574:
          31575:
          31576:
          31577:
          31578:
          31579:
          31580:
          31581:
          31582:
          31583:
          31584:
          31585:
          31586:
          31587:
          31588:
          31589:
          31590:
          31591:
          31592:
          31593:
          31594:
          31595:
          31596:
          31597:
          31598:
          31599:
          31600:
          31601:
          31602:
          31603:
          31604:
          31605:
          31606:
          31607:
          31608:
          31609:
          31610:
          31611:
          31612:
          31613:
          31614:
          31615:
          31616:
          31617:
          31618:
          31619:
          31620:
          31621:
          31622:
          31623:
          31624:
          31625:
          31626:
          31627:
          31628:
          31629:
          31630:
          31631:
          31632:
          31633:
          31634:
          31635:
          31636:
          31637:
          31638:
          31639:
          31640:
          31641:
          31642:
          31643:
          31644:
          31645:
          31646:
          31647:
          31648:
          31649:
          31650:
          31651:
          31652:
          31653:
          31654:
          31655:
          31656:
          31657:
          31658:
          31659:
          31660:
          31661:
          31662:
          31663:
          31664:
          31665:
          31666:
          31667:
          31668:
          31669:
          31670:
          31671:
          31672:
          31673:
          31674:
          31675:
          31676:
          31677:
          31678:
          31679:
          31680:
          31681:
          31682:
          31683:
          31684:
          31685:
          31686:
          31687:
          31688:
          31689:
          31690:
          31691:
          31692:
          31693:
          31694:
          31695:
          31696:
          31697:
          31698:
          31699:
          31700:
          31701:
          31702:
          31703:
          31704:
          31705:
          31706:
          31707:
          31708:
          31709:
          31710:
          31711:
          31712:
          31713:
          31714:
          31715:
          31716:
          31717:
          31718:
          31719:
          31720:
          31721:
          31722:
          31723:
          31724:
          31725:
          31726:
          31727:
          31728:
          31729:
          31730:
          31731:
          31732:
          31733:
          31734:
          31735:
          31736:
          31737:
          31738:
          31739:
          31740:
          31741:
          31742:
          31743:
          31744:
          31745:
          31746:
          31747:
          31748:
          31749:
          31750:
          31751:
          31752:
          31753:
          31754:
          31755:
          31756:
          31757:
          31758:
          31759:
          31760:
          31761:
          31762:
          31763:
          31764:
          31765:
          31766:
          31767:
          31768:
          31769:
          31770:
          31771:
          31772:
          31773:
          31774:
          31775:
          31776:
          31777:
          31778:
          31779:
          31780:
          31781:
          31782:
          31783:
          31784:
          31785:
         
```

0096 306
 0096 307 :++
 0096 308
 0096 309 : run down subroutine:
 0096 310
 0096 311 : checks ifab table for active files.
 0096 312 : if any found waits for any i/o activity to finish
 0096 313 : (doing a cancell i/o for non files-oriented devices)
 0096 314 : and then issues a \$close request.
 0096 315
 0096 316 : when all files run down performs a sanity check by seeing if all ifab
 0096 317 : table entries are also zero.
 0096 318
 0096 319 : inputs:
 0096 320 : r11 - impure area addr
 0096 321 : r8 - bit 0 set if indirect ppf run down
 0096 322 : ap - caller's arg list
 0096 323 : r7 - caller's mode
 0096 324
 0096 325 : outputs:
 0096 326 : returns only if noerror encountered.
 0096 327 : imp\$v_ppfindrd cleared
 0096 328 : r0 - R6, r9, r10 destroyed
 0096 329 :--
 0096 330
 0096 331 RUNDWN:
 0096 332 SSB #IMPSV_IORUNDOWN,(R11) ; set i/o rundown in progress flag
 009A 333 ; to sync with ast-driven rms
 009A 334 ; operations
 55 18 AB D0 009A 335 MOVL IMPSL_IFABTBL(R11),R5 ; get ifab table addr
 56 56 D4 009E 336 CLRL R6 ; build ifi value here
 54 52 85 D0 00A0 337 NXTSEG: MOVL (R5)+,R2 ; save addr next table seg in r2
 20 AB 3C 00A3 338 MOVZWL IMPSW_ENTPERSEG(R11),R4 ; get # entries/seg
 22 AB B5 00A7 339 NXTENT: TSTW IMPSW_NUM_IFABS(R11) ; any ifabs active?
 OF 13 00AA 340 BEGL CHKIRB ; branch if none
 56 D6 00AC 341 INCL R6 ; bump ifi
 5A 85 D0 00AE 342 MOVL (R5)+,R10 ; get ifab addr
 2F 12 00B1 343 BNEQ RDIFAB ; branch if one
 F1 54 F5 00B3 344 NXTSOB: SOBGTR R4,NXTENT ; keep scanning segment
 0086 345
 0086 346
 0086 347 : no more ifabs this segment, try next
 0086 348 :
 0086 349
 55 52 D0 0086 350 MOVL R2,R5 ; get next segment addr
 E5 12 0089 351 BNFE NXTSEG ; branch if one

00BB 353
 00BB 354 :
 00BB 355 : all ifabs have been run down now.
 00BB 356 :
 00BB 357 : unless this is indirect run down of ppf's.
 00BB 358 : check that all irabs are also gone.
 00BB 359 :
 00BB 360 :
 00BB 361 CHKIRB:
 00000008 00BB 362 .IF NE \$SRMSTEST&\$SRMS_TBUGCHK
 17 58 00 E4 00BB 363 BBSC #0,R8,30\$: branch if indirect run down
 55 1C AB D0 00BF 364 MOVL IMP\$L_IRABTBL(R11),R5 : get irab table addr
 54 52 85 D0 00C3 365 10\$: MOVL (R5)+,R2 : save addr next table seg.
 54 20 AB 3C 00C6 366 MOVZWL IMP\$W_ENTPERSEG(R11),R4 : get # entries/seg.
 85 D5 00CA 367 20\$: TSTL (R5)+ : entry zero?
 0D 12 00CC 368 BNEQ ERRBUG : branch if not
 F9 54 F5 00CE 369 SOBGTR R4,20\$: branch if more entries
 55 52 D0 00D1 370 MOVL R2,R5 : get next seg addr
 ED 12 00D4 371 BNEQ 10\$: branch if one
 00D6 372 30\$: CSB #IMP\$V_IORUNDOWN,(R11) : turn off rundown in progress flag
 00DA 373 .ENDC
 05 0CDA 374 RSB : all o.k.
 00DB 375 :
 00DB 376 :
 00DB 377 : close failed to zero ifab or irab table entry
 00DB 378 :
 00DB 379 ERRBUG: RMSTBUG FTLS_IORNDN

00E2 381
 00E2 382 : found an ifab. check for active and if so allow operation to finish
 00E2 383
 00E2 384 :
 00E2 385
 00E2 386 ASSUME IMPSW_RMSSTATUS EQ 0
 00E2 387 RDIFAB:
 2A 58 E8 00E2 388 BLBS R8, RDNET
 26 6A 20 E1 00E5 389 10\$: BBC #IFBSV_BUSY, (R10), RDNET ; branch if indirect ppf
 04 6A 0D E0 00E9 390 BBS #DEVSV_NET, (R10), 20\$; if not busy then check NETWORK
 3A 3A E1 00ED 391 BBC #IFBSV_RMS_STALL,- ; do cancel if busy & network operation
 30 6A 00EF 392 (R10), RDIRAB ; if this RMS thread is not currently
 00F1 393 staled then skip the cancel and wait
 00F1 394 :
 00F1 395 : allow function to finish
 00F1 396 : \note: this code should be modified to
 00F1 397 : properly run down read-ahead and write-behind
 00F1 398 : operations to unit record devices.\
 00F1 399 :
 00F1 400 20\$: SCANCEL_S IFBSW_CHNL(R10) ; cancel i/o (e.g. magtape create)
 FC 81 10 00FC 401 BSBB WAIT ; wait for an operation to finish
 A5 D5 00FE 402 TSTL -4(R5) ; ifab disappear? (close)
 B0 13 0101 403 BEQL NXTSOB ; branch if yes
 08 6A 20 E1 0103 404 BBC #IFBSV_BUSY, (R10), RDNET ; run down NETWORK if no longer busy
 E6 6A 0D E0 0107 405 BBS #DEVSV_NET, (R10), 20\$; but do cancel & wait again if busy &
 3A 3A E0 010B 406 BBS #IFBSV_RMS_STALL,- ; network operation or busy and the RMS
 E2 6A 010D 407 (R10), 20\$; thread is still stalled
 010F 408 :
 010F 409 :
 010F 410 : if the current operation is a network operation, and a special receive QIO
 010F 411 : has been posted but NOT received, a SCANCEL must always be done to flush
 010F 412 : this QIO. In file transfer mode it will be possible that a receive has been
 010F 413 : posted but no transfer operation is underway. therefore neither the IFAB nor
 010F 414 : the IRAB will be busy. if a SCANCEL isn't explicitly issued, when the SCLOSE
 010F 415 : is performed, the NETDRIVER will be unable to disconnect the logical link
 010F 416 : (because of the outstanding receive), and the process will hang.
 010F 417 :
 010F 418 :
 010F 419 :
 0E 6A 0D E1 010F 420 RDNET: BBC #DEVSV_NET, (R10), RDIRAB ; go run down IRABs if not network op
 50 3C AA D0 0113 421 MOVL IFBSV_NWA_PTR(R10), R0 ; obtain network work area address
 08 13 0117 422 BEQL RDIRAB ; skip check if not network work area
 03 E1 0119 423 BBC #NWASV_RCVQIO,- ; if a special receive QIO has not been
 04 60 011B 424 (R0), RDIRAB ; posted go run down the IRABs, but if
 04 E1 011D 425 BBC #NWASV_RCVAST,- ; one has and it hasn't been received
 29 60 011F 426 (R0), CANCEL ; then go issue the cancel
 0121 427 :
 0121 428 : run down irabs
 0121 429 :
 0121 430 :
 0121 431 :
 0121 432 RDIRAB:
 59 59 5A D0 0121 433 MOVL R10, R9 ; copy ifab addr
 1C A9 D0 0124 434 10\$: MOVL IRBSL_IRAB_LNK(R9), R9 ; get next irab
 30 13 0128 435 BEQL QUIET ; branch if none
 03 58 E8 012A 436 BLBS R8, 12\$; don't release locks if indirect PPF
 012D 437 : rundown

FED0'	30	012D	438		BSBW	RMSUNLOCKALL	: kill all record locks, including : outstanding waits.
FO 69	20	E1	0130	439	BBC	#IRBSV_BUSY,(R9),10\$: branch if idle
04 6A	0D	E0	0134	440	BBS	#DEVSV_NET,(R10),15\$: do cancel if busy & network operation
3A	E1	0138	441		BBC	#IRBSV_RMS_STALL,-	: if this RMS thread is not currently
EB 69		013A	442			(R9),10\$: stalled then skip the cancel and wait
		013C	443				
		013C	444				
03 69	07 6B	E8	013C	445	BLBS	(R11), 20\$: branch if image i/o segment
	22	E0	013F	446	BBS	#IRBSV_PPF_IMAGE,(R9),20\$	
		0143	447				
		0143	448				
		0143	449			: branch if indirect i/o	
		0143	450				
		0143	451				
DE 58	E9	0143	452		BLBC	R8,10\$: branch if only indirect ppts
		0146	453				: to be run down
1C	E0	0146	454	20\$:	BBS	#DEVSV_RND,-	
6A		0148	455			IFBSL_PRIM_DEV(R10),-	: no need to do a cancel if this is
DB		0149	456			NOCANCEL	: a disk operation, just go wait
		014A	457				
		014A	458	CANCEL: SCANCEL_S		IFBSW_CHNL(R10)	: cancel i/o
		0155	459				
		0155	460	NOCANCEL:			
FF27	30	0155	461		BSBW	WAIT	: wait for all ASTs to be delivered
C7	11	0158	462		BRB	RDIRAB	: start from top of irab
		015A	463				: chain again (could
		015A	464				: have been disconnect)

015A 466
 015A 467 :
 015A 468 : all activity ceased for this file.
 015A 469 : force a close by constructing a fab and calling close.
 015A 470 :
 015A 471 :
 78 AA D5 015A 472 QUIET: TSTL IFBSL_SF\$B_PTR(R10) ; is it a shared file?
 OC 12 015D 473 BNEQ \$S ; yes, go close it
 35 6A 30 E1 015F 474 BBC #IFBSV_WRTACC,(R10),NOERR ; branch if not write access
 31 6A 03 E1 0163 475 BBC #DEVSV_DIR,IFBSL_PRIM_DEV(R10),NOERR
 2D 6A 25 E1 0167 476 BBC #IFBSV_ACCESSED,(R10),NOERR ; branch if file not accessed
 50 04 AC D0 016B 477 5\$: MOVL 4(AP),R0 ; get descriptor addr
 016F 478 IFNORD #8,(R0),NOERR1,R7
 1C 60 B1 0175 479 CMPW (R0),#28 ; at least 22 bytes long?
 17 1F 0178 480 BLSSU NOERR1
 53 04 A0 D0 017A 481 MOVL 4(R0),R3 ; get buffer address
 59 5A D0 017E 482 MOVL R10,R9 ; ifab to right register
 0069 30 0181 483 IFNOWRT #22,(R3),NOERR1,R7 ; branch if buffer not writable
 0187 484 BSBW GETDVIFID ; go fill buffer with dvi and fid
 018A 485 RMSERR CCF,R3 ; get set for close failure
 OA 11 018F 486 BRB CLOSE
 0191 487 :
 0191 488 NOERR1: RMSERR IAL,R3 ; if close failure, return ial
 03 11 0196 489 BRB CLOSE
 0198 490 :
 0198 491 NOERR: RMSSUC SUC,R3 ; can't fail
 0198 492 :
 SE B0 AE DE 0198 493 CLOSE: MOVAL -FABSC_BLN(SP),SP ; create fab on stack
 5003 8F B0 019F 494 MOVW #FABSC_BID+<FABSC_BLN 28>,-(SP) ; fab block id and length
 6E 01A3 495 :
 02 AE 56 B0 01A4 496 MOVW R6,FABSWIFI(SP) ; ifi
 0B 6B E8 01A8 497 BLBS (R11),10\$; branch if fio seg
 04 58 E9 01AB 498 SSB #15+<FABSWIFI*8>,(SP) ; set pio flag
 01AF 499 BLBC R8,10\$; branch if direct access
 01B2 500 SSB #FABSV_PPF_IND+<FABSWIFI*8>,-(SP) ; else make indirect ifi
 01B2 501 :
 14 AE 00 3C BB 01B6 502 10\$: PUSHR #^M<R2,R3,R4,R5> ; save regs
 00 6E 00 2C 01B8 503 MOVC5 #0,(SP),#0,- ; zero remainder of fab
 004C 8F 01BC 504 POPR #FABSC_BLN-4,4+<4+4>(SP) ; restore r5
 3C BA 01C1 505 BSBW ENBAST
 FE9F 30 01C3 506 PUSHL SP ; addr of fab
 5E DD 01C6 507 CALLS #1,2#SYS\$CLOSE ; close it
 00000000'9F 01 FB 01C8 508 BSBW INHAST
 FEAS 30 01CF 509 ADDL #FABSC_BLN,SP ; 'pop' fab
 5E 00000050 8F C0 01D2 510 BLBS R8,15\$; omit check if indirect ppf
 05 58 E8 01D9 511 TSTL -4(R5) ; did ifab go away?
 FC AS D5 01DC 512 BNEQ ERRBUG_BR ; branch if not
 OF 12 01DF 513 BLBC R0,30\$; branch on error
 03 50 F9 01E1 514 15\$: BRW NXFSOB ; get next ifab
 FECC 31 01E4 515 20\$: MOVL R3,R0 ; get saved error code
 50 53 D0 01E7 516 30\$: BLBS R0,20\$; no problem if not
 F7 50 E8 01EA 517 : write-accessed file
 FE5E 31 01ED 518 : return error to caller
 FEE8 31 01F0 519 :
 FEE8 31 01F0 520 ERRBUG_BR: EXIT
 01F3 521 :
 FEE8 31 01F0 522 ERRBUG : extended branch

01F3 524 :++
01F3 525 : GETDEVIFID -- Get Device ID and File ID.
01F3 526 :
01F3 527 : This routine returns the counted device name string,
01F3 528 : as well as the file id for the file open on the channel.
01F3 529 :
01F3 530 : Calling Sequence:
01F3 531 :
01F3 532 : BSBW GETDVIFID
01F3 533 :
01F3 534 : Input Parameters:
01F3 535 :
01F3 536 : R9 IFAB address
01F3 537 : R3 address of 22-byte buffer to return device name string
01F3 538 : IFBSW_CHNL channel #
01F3 539 :
01F3 540 : Implicit Inputs:
01F3 541 :
01F3 542 : none
01F3 543 :
01F3 544 : Output Parameters:
01F3 545 :
01F3 546 : R0,R1,R3 destroyed
01F3 547 :
01F3 548 : Implicit Outputs:
01F3 549 :
01F3 550 : The counted ascii string for the device name is moved
01F3 551 : to the buffer provided, followed by the file id starting 16 bytes
01F3 552 : from the start of the buffer.
01F3 553 :
01F3 554 :--
01F3 555 :--

0434 BF 88 01F3 557 GETDVIFID:
 53 DD 01F7 558 PUSHR #^M<R2,R4,R5,R10> ; Save regs.
 SA 38 A9 01F9 559 PUSHL R3 ; Save R3.
 83 0190 CA 90 01FD 560 MOVL IFBSL_FWA_PTR(R9),R10 ; Get FWA into R10.
 0190 CA 28 0202 561 MOVB FWASQ_SHRFIL(R10),(R3)+ ; Move size of buffer id into first byte of
 0194 DA 0206 562 MOVC3 FWASQ_SHRFIL(R10)- ; Move device id name into buffer
 63 0209 563 @FWASQ_SHRFIL+4(R10),-
 53 BED0 020A 564 (R3)
 020D 565 POPL R3 ; Restore R3.
 020D 566
 020D 567
 020D 568 :
 020D 569 : Now get the file ID from the FWA.
 020D 570 : R3 = address of the specified output buffer
 020D 571 :
 020D 572 :
 10 A3 D4 020D 573 CLRL 16(R3) ; Clear FID field in buffer.
 14 A3 B4 0210 574 CLRW 20(R3)
 07 69 05 E0 0213 575 BBS #DEV\$V_SQD,(R9),10\$; branch if magtape (no FCB)
 06 28 0217 576 MOVC3 #6,- ; Move FID to buffer.
 01F8 CA 0219 577 FWAST_FIBBUF+FIBSW_FID(R10),-
 10 A3 Q21C 578 16(R3)
 0434 BF BA 021E 579 10\$: POPR #^M<R2,R4,R5,R10> ; Restore regs.
 05 0222 580 RSB
 0223 581
 0223 582 .END

\$\$.PSECT_EP	= 00000000	NWASB_DSTYPE	000000C4
\$SRMSTEST	= 0000001A	NWASB_RF	000000C7
\$SRMS_PBUGCHK	= 00000010	NWASB_RMS_RAC	000000C8
\$SRMS_TBUGCHK	= 00000008	NWASC_BLN	00000800
\$SRMS_UMODE	= 00000004	NWASK_BLN	00000800
CANCEL	0000014A R 01	NWASL_ALLXABADR	00000100
CHKIRB	000000B8 R 01	NWASL_DATXABADR	00000104
CLOSE	0000019B R 01	NWASL_DEV	000000C0
DEV\$V_DIR	= 00000003	NWASL_FHCXABADR	00000108
DEV\$V_NET	= 0000000D	NWASL_KEYXABADR	0000010C
DEV\$V_RND	= 0000001C	NWASL_MSG_MASK	000000D4
DEV\$V_SQD	= 00000005	NWASL_PRORABADR	00000110
ENBAST	00000065 R 01	NWASL_RDTXABADR	00000114
ERRBUG	000000DB R 01	NWASL_SAVE_FLGS	00000128
ERRBUG_BR	000001F0 R 01	NWASL_SUMXABADR	00000118
ERRBUSY	0000005B R 01	NWASL_THREAD	000000FC
EXIT	0000004E R 01	NWASL_XLTATTR	00000238
FABSC_BID	= 00000003	NWASL_XLTBUFFLG	0000022C
FABSC_BLN	= 00000050	NWASL_XLTCNT	00000228
FABSV_PPF_IND	= 0000000E	NWASL_XLTMAXINDX	00000234
FABSWIFI	= 00000002	NWASL_XLTSIZ	00000230
FIBSW_FID	= 00000004	NWASQ_ACS	00000244
FTLS_IORNDN	= FFFFFFE	NWASQ_BIGBUF	00000170
FWASQ_SHRFIL	= 00000190	NWASQ_BLD	000000F0
FWAST_FIBBUF	= 000001F4	NWASQ_FLG	00000000
GETDVIFID	000001F3 R 01	NWASQ_INODE	0000025C
IFBSL_FWA_PTR	= 00000038	NWASQ_IOSB	000000D8
IFBSL_NWA_PTR	= 0000003C	NWASQ_LNODE	00000160
IFBSL_PRIM_DEV	= 00000000	NWASQ_LOGNAME	0000023C
IFBSL_SFSP_PTR	= 00000078	NWASQ_NCB	00000264
IFBSV_ACCESED	= 00000025	NWASQ_RCV	000000E0
IFBSV_BUSY	= 00000020	NWASQ_SAVE_DESC	00000120
IFBSV_RMS_STALL	= 0000003A	NWASQ_XLTBUF1	0000024C
IFBSV_WRTACC	= 00000030	NWASQ_XLTBUF2	00000254
IFBSW_CHNL	= 00000020	NWASQ_XMT	000000E8
IMPSC_IOREFN	= 0000001E	NWAST_ACSBUF	0000026C
IMPSL_IFABTBL	= 00000018	NWAST_AUXBUF	000005E0
IMPSL_IRABTBL	= 0000001C	NWAST_DAP	00000000
IMPSV_IORUNDOWN	= 00000004	NWAST_INODEBUF	000004AC
IMPSW_ENTPERSEG	= 00000020	NWAST_ITM_ATTR	00000200
IMPSW_NUM_IFABS	= 00000022	NWAST_ITM_END	00000224
IMPSW_RMSSTATUS	= 00000009	NWAST_ITM_LST	00000200
INHAST	00000077 R 01	NWAST_ITM_MAXINDX	00000218
IRBSL_IRAB_LNK	= 0000001C	NWAST_ITM_STRING	0000020C
IRBSV_BUSY	= 00000020	NWAST_NCBBUF	0000052C
IRBSV_PPF_IMAGE	= 00000022	NWAST_NODEBUF	00000169
IRBSV_RMS_STALL	= 0000003A	NWAST_RCVBUF	000001A0
NOCANCEL	00000155 R 01	NWAST_SCAN	00000100
NOERR	00000198 R 01	NWAST_TEMP	00000120
NOERR1	00000191 R 01	NWAST_XLTBUF1	000002AC
NWASB_ALLXABCNT	0000011C	NWAST_XLTBUF2	000003AC
NWASB_DAP_RAC	000000C9	NWAST_XMTBUF	000003C0
NWASB_FILESYS	000000C5	NWASV_RCVAST	= 00000004
NWASB_KEYXABCNT	0000011D	NWASV_RCVQIO	= 00000003
NWASB_NETSTRSIZ	0000016F	NWASW_BUILD	000000D2
NWASB_NODBUFSIZ	00000168	NWASW_DAPBUFSIZ	000000CA
NWASB_ORG	000000C6	NWASW_DIR_OFF	000000CC

RMSORNDWN
Symbol table

RMS IO RUN DOWN

J 14

16-SEP-1984 01:29:13 VAX/VMS Macro V04-00
14-SEP-1984 22:32:57 [RMS.SRC]RMSORNDWN.MAR;2

Page 17
(15)

NWASH DISPLAY
NWASH FIL OFF
NWASH JNLXABJOP
NXTENT
NXTSEG
NXTSOB
PIO\$A TRACE
PIO\$GW_IIOIMPA
PIO\$GW-PIOIMPA
PIO\$GW-STATUS
PIO\$V_INHAST
PSLSC_USER
PSLSS_PRVMOD
PSL\$V_PRVMOD
QUIET
RDIFAB
RDIRAB
RDNET
RMSBUG
RMSLAST CHANCE
RMSRU_UNLOCK
RMSUNLOCKALL
RMSSRMSRUNDWN
RMSS_BUSY
RMSS_CCF
RMSSIAL
RMSABORT
RUNDWN
SY\$CANCEL
SY\$CLOSE
SY\$CLREF
SY\$SETAST
SY\$WAITFR
TPTSL_RUNDWN
WAIT
XITSUC

00000000
000000CE
0000011E
000000A7 R 01
000000A0 R 01
000000B3 R 01
***** X 01
***** X 01
***** X 01
= 00000000
= 00000003
= 00000002
= 00000016
0000015A R 01
000000E2 R 01
00000121 R 01
0000010F R 01
***** X 01
***** X 01
***** X 01
***** X 01
= FFFFFFFE RG 01
= 0001848C
= 0001C0DC
= 0001854C
00000055 R 01
00000096 R 01
***** GX 01
***** X 01
***** GX 01
***** GX 01
***** GX 01
***** X 01
0000007F R 01
0000004B R 01

+-----+
! Psect synopsis !
+-----+

PSECT name

. ABS .
RMSRMS
SAB\$

Allocation PSECT No. Attributes

00000000 (0.) 00 (0.) NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
00000223 (547.) 01 (1.) PIC USR CON REL GBL NOSHR EXE RD NOWRT NOVEC BYTE
00000800 (2048.) 02 (2.) NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE

+-----+
! Performance indicators !
+-----+

Phase

Initialization
Command processing
Pass 1
Symbol table sort

Page faults CPU Time Elapsed Time

36 00:00:00.08 00:00:00.84
142 00:00:00.72 00:00:04.05
417 00:00:16.06 00:00:41.85
0 00:00:02.27 00:00:04.14

RMS
VO

RMSORNDWN
VAX-11 Macro Run Statistics

RMS IO RUN DOWN

K 14

16-SEP-1984 01:29:13 VAX/VMS Macro V04-00
14-SEP-1984 22:32:57 [RMS.SRC]RMSORNDWN.MAR;2

Page 18
(15)

Pass 2	109	00:00:02.95	00:00:06.37
Symbol table output	19	00:00:00.18	00:00:00.26
Psect synopsis output	1	00:00:00.04	00:00:00.04
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	726	00:00:22.31	00:00:57.63

The working set limit was 1650 pages.

86675 bytes (170 pages) of virtual memory were used to buffer the intermediate code.
There were 90 pages of symbol table space allocated to hold 1649 non-local and 23 local symbols.
582 source lines were read in Pass 1, producing 14 object records in Pass 2.
37 pages of virtual memory were used to define 36 macros.

+-----+
! Macro library statistics !
+-----+

Macro library name

Macros defined

\$255\$DUA28:[RMS.OBJ]RMS.MLB;1	18
\$255\$DUA28:[SYS.OBJ]LIB.MLB;1	3
\$255\$DUA28:[SYSLIB]STARLET.MLB;2	11
TOTALS (all libraries)	32

1838 GETS were required to define 32 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LISS:RMSORNDWN/OBJ=OBJ\$:RMSORNDWN MSRC\$:RMSORNDWN/UPDATE=(ENH\$:RMSORNDWN)+EXECML\$/LIB+LIB\$:RMS/LIB

0330 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

